

# PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2000-284962

(43)Date of publication of application : 13.10.2000

(51)Int.Cl.

G06F 9/318

G06F 9/30

(21)Application number : 11-094060

(71)Applicant : SEIKO EPSON CORP

(22)Date of filing : 31.03.1999

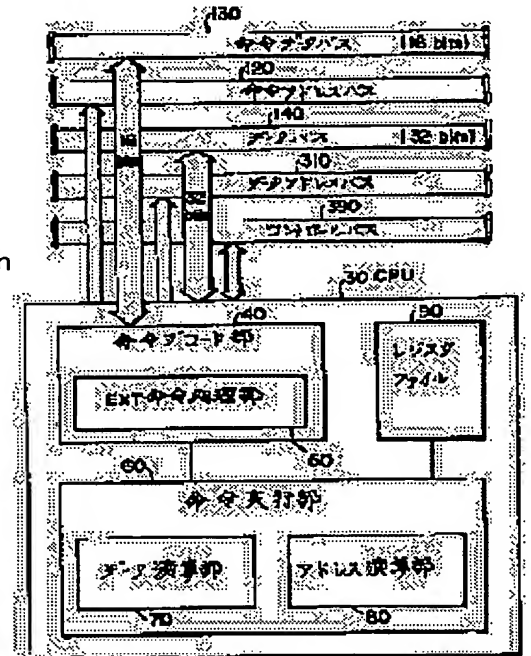
(72)Inventor : MATSUOKA HIROKI

## (54) MICROCOMPUTER

### (57)Abstract:

**PROBLEM TO BE SOLVED:** To provide a function for extending operation contents for executing an operation which cannot be described in a short instruction code.

**SOLUTION:** This microcomputer operates processing necessary for decoding an inputted instruction code, and for executing the instruction, and this microcomputer is provided with an instruction decode part 40, a register file 90, and an instruction executing part 60 which executes the instruction based on the operation contents of the instruction analyzed by the instruction decode part 40. The instruction decode part 40 includes an ext instruction processing part 50 which operates processing for extending an operation according to a prefix instruction. In this case, the prefix instruction is combined with a target instruction so that it is possible to perform access to a wider address space with two instructions by using register data in wide width as an offset value. Thus, an instruction code size can be reduced, and the number of execution cycles can be reduced. Also, an inter-register three operand instruction can be realized while this is difficult in a CPU in a type that the length of an instruction is short.



## LEGAL STATUS

[Date of request for examination]

08.04.2003

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of

rejection]

[Date of requesting appeal against examiner's  
decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

**\* NOTICES \***

Japan Patent Office is not responsible for any damages caused by the use of this translation.

1.This document has been translated by computer. So the translation may not reflect the original precisely.

2.\*\*\*\* shows the word which can not be translated.

3.In the drawings, any words are not translated.

---

**CLAIMS**

---

[Claim(s)]

[Claim 1] An instruction-execution means is a microcomputer characterized by executing this target instruction from the contents of operation extended with said target instruction expansion means including a target instruction expansion means to extend and interpret the contents of operation of this target instruction when an instruction code analysis means inputs a target instruction after a prefix instruction input.

[Claim 2] Said target instruction expansion means according to claim 1 is a microcomputer characterized by extending the contents of operation of this target instruction to 3 operand instructions, and interpreting them using the contents of this register contained in two operands and prefix instructions of this target instruction when the predetermined target instruction which is 2 operand instructions is inputted after a prefix instruction input.

[Claim 3] Said target instruction expansion means according to claim 1 is the microcomputer characterized by to extend the contents of operation so that the displacement of the address stored in the register specified as the instruction code of a target instruction creates based on the contents of the register of the register number contained in the instruction code of a prefix instruction and it may perform using this displacement, when the predetermined target instruction which has a register assignment value after a prefix instruction input is inputted.

---

[Translation done.]

**\* NOTICES \***

Japan Patent Office is not responsible for any damages caused by the use of this translation.

1.This document has been translated by computer. So the translation may not reflect the original precisely.

2.\*\*\* shows the word which can not be translated.

3.In the drawings, any words are not translated.

---

**DETAILED DESCRIPTION**

---

[Detailed Description of the Invention]

[0001]

[Field of the Invention] This invention relates to the electronic equipment constituted using an information processing circuit and semiconductor integrated circuit equipment, the microcomputer that contains said information processing circuit, and this microcomputer.

[0002]

[Description of the Prior Art] Conventionally, in the microcomputer of a RISC method which can process 32-bit data, the instruction code fixed to 32-bit width of face is used. The reason is that the

time amount which decoding of an instruction takes can be shortened compared with the case where the instruction code of variable-length bit width of face is used, and it can make the circuit scale of a microcomputer small if the instruction code of fixed-length bit width of face is used.

[0003] However, also in a 32-bit microcomputer, there is also much instruction code which is not needed especially 32 bits. Therefore, if 32 bits describes the instruction code of all instructions, the instructions which a redundant part produces in instruction code will increase in number, and the utilization ratio of memory will worsen.

[0004] In such a case, it is also possible to execute an instruction, carrying out the logic reduction of the redundant instruction code, and decoding to the original instruction inside a microcomputer. However, there was a problem that a control circuit became complicated, by such method.

[0005]

[Problem(s) to be Solved by the Invention] then, this invention — the person was performing examination about the microcomputer which processes the fixed-length instruction code of bit width of face shorter than the bit width of face of the data which can be processed in order to raise the utilization ratio of memory, without complicating a control circuit. However, if 32 bit fixed length's instruction code is only made into a 16-bit fixed length, for example, the following troubles will arise. For example, it is difficult to secure the field of the operand of 3 operand instructions in 16-bit instruction code. Therefore, the problem how to process the operation that description is difficult, by short instruction code to this appearance arises. In order to perform operation which cannot be described by short instruction code in order to solve the above-mentioned problem, the function which extends the contents of operation is needed.

[0006] Moreover, other purposes of this invention are offering the information processing circuit which has the function which extends the contents of operation, semiconductor integrated circuit equipment, and a microcomputer, without complicating a control circuit.

[0007]

[Means for Solving the Problem] The instruction processed with the microcomputer of this invention can be divided into a usual instruction and a usual prefix instruction. About a predetermined instruction, it can become the target of a prefix instruction among the usual instructions (a target \*\*\*\*\* instruction of a prefix instruction is called target instruction). A prefix instruction has the function which extends the target instruction function, in case [ its ] it does not perform in an instruction-execution means but a target instruction of consecutiveness is executed, if independent.

[0008] In order to attain said purpose, invention of claim 1 is the information-processing circuit where a given target instruction and the prefix instruction for extending this target instruction function are inputted, and inputs said target instruction and a prefix instruction, and it is characterized by to include an instruction code analysis means analyze the contents of operation of this instruction code, and an instruction-execution means execute an instruction based on the contents of operation which said instruction code analysis means analyzed.

[0009] An instruction code analysis means decodes the inputted instruction code, and performs processings required in order that an instruction-execution means may execute an instruction, such as calculating the address of the storage means which is set as the object of an instruction and by which the data storage is carried out.

[0010] When said instruction code analysis means inputs a target instruction after a prefix instruction input, said instruction-execution means is characterized by executing this target instruction from the contents of operation extended with said target instruction expansion means including a target instruction expansion means to extend and interpret the contents of operation of this target instruction.

[0011] If it is made this appearance, the function extended to the target instruction can be realized by performing a target instruction combining a prefix instruction. For this reason, the number of instructions can be reduced and it becomes possible to reduce the number of bits further used for instruction code.

[0012] Moreover, in instruction code with little number of bits, it also becomes possible to realize the function that implementation is difficult, by combining a prefix instruction and a target instruction. For this reason, the utilization ratio of the memory by compaction of instruction code is improvable.

[0013] Invention of claim 2 is characterized by to extend the contents of operation of this target instruction to 3 operand instructions, and to interpret them using the data of the register of a number with which said target instruction expansion means was included in two operands and prefix instructions of this target instruction when the predetermined target instruction which is 2 operand instructions was inputted after a prefix instruction input, in claim 1.

[0014] If it carries out such, by performing a target instruction of 2 operand instructions combining a prefix instruction, the actuation function of this target instruction can be extended to 3 operand instructions, and can be performed. Therefore, even if it uses the short instruction code which has only a description field by two operands, 3 operand instructions are realizable. For this reason, it becomes possible to reduce the number of bits used for the number of instructions, and instruction code.

[0015] Invention of claim 3 is characterized by to extend the contents of operation so that the data of the register of the number contained in the instruction code of a prefix instruction when said target instruction expansion means inputted the predetermined target instruction which has a register assignment value after a prefix instruction input use in claim 1 as displacement of the address stored in the register specified as the instruction code of a target instruction and it may perform .

[0016] If it does in this way, by using a prefix instruction, without being limited to the number of bits of an operand, addition of displacement is attained and can increase displacement to a part for the bit of the data width of face of a register.

[0017] Moreover, by processing fixed-length short instruction code, the microcomputer of a RISC method shortens the time amount which decoding of an instruction takes, and aims at making the circuit scale of a microcomputer small. Therefore, according to this invention, the microcomputer of the RISC method which can realize these purposes easily can be offered.

[0018]

[Embodiment of the Invention] Hereafter, the gestalt of operation of this invention is explained based on a drawing.

[0019] 1. Functional description drawing 1 of CPU is the functional block diagram of CPU of the gestalt of operation of this invention. Although this CPU30 treats the data of 32-bit width of face, it is constituted so that the instruction code which is 16 bits may be processed.

[0020] A book CPU 30 contains the instruction decoding section 40, the instruction-execution section 60, and a register file 90. And this CPU performs an exchange of the exterior and a signal through the 16-bit instruction data bus 130, the instruction address bus 120 for an instruction data access, the 32-bit data bus 140, the data address bus 310 for a data access, and the control bus 390 for a control signal.

[0021] Said instruction decoding section 40 is also performing processing required in order to decode the inputted instruction code and to execute this instruction, and contains the ext instruction-processing section 50 which is the characteristic function of this invention. The register file 90 has the register used by CPUs, such as 16 general-purpose registers of general-purpose registers R0-R15, a program counter (PC), a processor status register (PSR), a stack pointer (SP), an arithmetic low register (ALR), and an arithmetic high register (AHR). The instruction-execution section 60 executes this instruction based on the contents of operation of the instruction which said instruction decoding section 40 analyzed, and contains the data operation part 70 which calculates data, and the address-arithmetic section 80 which performs the operation of the address.

[0022] Since the data size of the hardware inside this CPU is 32 bits, all of the arithmetic logical operation performed in the data operation part 70 of said instruction-execution section 60 or the address-arithmetic section 80, the number of bits, a data transfer of a register of said register file, etc. become 32 bits. However, the instruction code which said CPU30 processes is a 16-bit fixed length.

Usually, instruction code consists of operands of the class code and operation code which define the fundamental operation of this instruction, a source register (rs) and a destination register (rd), and an immediate. When using 16-bit instruction code, the number of bits of the immediate which can be used becomes small, and it becomes impossible for this reason, for a source register to specify only one.

[0023] So, the instruction of 16 bits of fixed lengths called an ext instruction is newly prepared, and it constitutes from a gestalt of this operation so that a register number can be specified. Moreover, by using an ext instruction, it constitutes from instruction code of short bit width of face called 16 bits of fixed lengths so that operation which cannot be described may be made possible. Although an ext instruction does not perform its activation in CPU if independent at all, it is the prefix instruction which is an instruction which extends the target instruction function by Lycium chinense just before a target instruction. In addition, the instruction set as the object of the expansion by this prefix instruction is called target instruction. The configuration of the instruction code of an ext instruction has the 8-bit register appointed field 403 in the bit 0 from the bit 15 to the bit 13 from the field 502 which specifies a 5-bit operation code, and the bit 7 from the bit 12 to the class appointed field 501 of a triplet, and the bit 8, as shown in drawing 5. The code which shows one of general-purpose registers is stored in the register appointed field 403.

[0024] Therefore, with the gestalt of this operation, it is characterized by extending the operation of a target instruction of consecutiveness using the value of the general-purpose register specified with a prefix instruction. The ext instruction-processing section 50 processes such an ext instruction, and it mentions later for details.

[0025] 2. Take for an example the semiconductor integrated circuit built in the configuration microcomputer of a semiconductor integrated circuit, and explain an example and activity of circuitry for realizing the characteristic function of this invention.

[0026] Drawing 2 illustrates a part required for explanation of the configuration of the semiconductor integrated circuit built in the microcomputer. This semiconductor integrated circuit 100 contains ROM110 and RAM320 with CPU30.

[0027] Said CPU30 performs an exchange of the exterior and a signal through the data address bus 310 for the data bus 140 of 120 or 32 bits of instruction address buses for said ROM110, said RAM320, the 16-bit instruction data bus 130, and this instruction data access, and this data access.

[0028] The 16-bit instruction code which described the processing which CPU30 is made to perform is memorized by said ROM110, and it functions on it as an instruction code storage means. And the instruction code of said prefix instruction or the instruction used as a target as well as the instruction code of the usual instruction is memorized by this ROM110.

[0029] Said RAM320 has memorized the data used for activation.

[0030] Said CPU30 contains an instruction register 150, the instruction decoding circuit 160, the EXT register 172 and a register file 90, ALU190, and a program counter PC 330 and the address-arithmetic machine 340.

[0031] An instruction register 150 stores the instruction code inputted through the instruction data bus 130 from said ROM110. The instruction decoding circuit 160 decodes the instruction code stored in the instruction register 150, and outputs the control signal 161 required for activation. Moreover, the general-purpose register value in instruction code is started, and it holds to an EXT register if needed. Although the address-arithmetic machine 340 calculates four operations etc., it takes up only the function which carries out an add operation for convenience, and this example explains it.

[0032] The address of the instruction under current activation is stored in the program counter PC 330.

[0033] ALU190 performs arithmetic operation, logical operation, and shift operation to the data of the value stored in the register of said extended immediate and register file 90, or said RAM320. That is, ALU190 functions as data operation part 70 of the instruction-execution section 40 shown in drawing 1.

[0034] The target instruction expansion circuit 170 extends the operation of this target instruction from the target instruction which the instruction decoding circuit 160 judged, and the instruction EXT register

172. That is, the instruction decoding circuit 160, the target instruction expansion circuit 170, and the EXT register 172 function as the instruction decoding section 40 shown in drawing 1 , and the ext instruction-processing section 50.

[0035] 3. Explain a means to hold the number of the general-purpose register contained in the instruction code of an example \*\*\*\* prefix instruction of the escape of the target instruction using an ext instruction (prefix instruction). As a means, the instruction decoding circuit's 160 detection of an ext instruction saves the data of the corresponding register at the EXT register 172.

[0036] Next, a means to change the operation of a target instruction using the data of said EXT register is explained.

[0037] Drawing 3 is the timing-chart Fig. having shown relation with the control signal which the instruction code inputted and said instruction decoding circuit generate.

[0038] Said control signal which said instruction decoding circuit generates is with the ext signal 330 and the ext\_low signal 340. 310 of drawing 3 expresses the instruction code inputted. Moreover, a clock signal 320 is a signal which is generated by the clock signal generator which is not illustrated in drawing 2 , or is inputted from a clock input terminal. This clock signal 520 is used in order to take the synchronization of the various actuation in CPU. For example, synchronizing with the standup of this clock signal 320, the instruction address is outputted to the instruction address bus 120. Moreover, based on said instruction address, instruction code is read from ROM110 to every 1 of this clock signal 320 period (one machine cycle), and it is held at an instruction register 150. And the operation according to the read instruction code is completed within 1 machine cycle.

[0039] The ext signal 330 is a signal used as '1', when an ext instruction is inputted. That is, said instruction decoding circuit sets the ext signal 330 to '1', when the inputted instruction is an ext instruction, and when the inputted instruction is the usual instruction, it operates so that the ext signal 330 may be set to '0'.

[0040] The ext\_low signal 340 is a signal outputted according to the condition of the ext signal 340. That is, when the ext signal 330 is '1', said instruction decoding circuit takes a synchronization in the standup of the following clock signal, sets the ext\_low signal 340 to '1', takes a synchronization in the standup of the following clock signal, and sets the ext\_low signal 340 to '0'.

[0041] When the ext signal 330 is '0'ext\_low signal 550='0', it is in the condition of processing the usual instruction, and, as for the target instruction escape circuit 170, operation is not changed in this case.

[0042] When the ext signal 330 is '1', CPU is in the condition of processing the ext instruction which is a prefix instruction. At this time, the data of the register which an ext instruction specifies are saved at the EXT register 172. In this case, the target instruction escape circuit 170 does not change operation.

[0043] The ext signal 330 is at '0'. In ext\_low signal 550='1', it is in the condition of processing the usual instruction which is a target instruction after an ext instruction. In this case, usage operation is changed for the data of the assignment register contained in the instruction code of an ext instruction of the point currently held at the EXT register 172.

[0044] 4. Explain the typical example of the contents escape of operation of the target instruction by prefix instruction, next the gestalt of typical implementation of the contents escape of operation of the target instruction by the prefix instruction of this invention taking the case of the case of an instruction (henceforth Type 1 instruction for convenience) of the type which calculates using 2 general-purpose registers.

[0045] First, the configuration of the instruction code of Type 1 instruction is explained. Drawing 5 is drawing having shown the bit field of the instruction code of this Type 1 instruction.

[0046] As shown in this drawing, the instruction code of Type 1 instruction has the 4-bit register appointed field 404 in the bit 4 from the bit 15 to the bit 13 from the field 402 which specifies a 5-bit operation code, and the bit 7 from the bit 3 to the 4-bit register appointed field 403 and the bit 0 from the bit 12 to the class appointed field 401 of a triplet, and the bit 8.

[0047] Moreover, the code which shows one of general-purpose registers (rs) and (rd), respectively is

stored in said register appointed fields 403 and 404. the register source is meant in rs and a register DISUTE nation is meant in rd — it is. Type 1 instructions are 2 operand instructions which perform the operation shown so much in the data stored in the general-purpose register (rs), and the data stored in the general-purpose register (rd) by the operation code, and perform actuation which writes a result in a general-purpose register (rd).

[0048] Next, actuation of CPU when Type 1 instruction is executed independently is explained. In drawing 2, Type 1 instruction is first inputted into an instruction register 150 through the instruction data bus 130 from ROM110. And it decodes in the instruction decoding circuit 160, and the operation according to the contents is performed. In Type 1 instruction, the data stored in the general-purpose register (rd) specified by instruction code are inputted into ALU190 through a data bus 182 from a register file 90. Moreover, the data stored in the general-purpose register (rs) specified by instruction code are inputted into ALU190 through a data bus 184 from a register file 90. And ALU190 performs the operation shown by the operation code of Type 1 instruction, and stores the result of an operation in the general-purpose register (rd) of a register file 90 through a data bus 192.

[0049] Said Type 1 instruction can also be executed independently and it is also possible to perform combining the ext instruction of just before \*\*.

[0050] When executing Type 1 instruction independently, the formula showing the operation in the case of performing combining an ext instruction is as follows.

[0051] (A) Operation  $rd=rs$  of Type 1 instruction Operation  $rd=rs$  of op.rd(B) EXT instruction + type 1 instruction An op.EXT register type (A) is a formula showing operation when Type 1 instruction is executed independently. Since original operation of Type 1 instruction is performed as shown in this formula when performing independently, the data of an EXT register are not used. In addition, it is as having mentioned above about the contents of activation when Type 1 instruction is executed independently.

[0052] A formula (B) is a formula showing operation when Type 1 instruction is executed combining the last ext instruction. As shown in this formula, when Type 1 instruction is executed combining the last ext instruction, the contents of operation extend the result of an operation of the data of an EXT register, and the contents of the general-purpose register (rs) so that it may write in a general-purpose register (rd).

[0053] The contents of activation when Type 1 instruction is executed combining an ext instruction are explained. In drawing 2, Type 1 instruction is first inputted into an instruction register 150 through the instruction data bus 130 from ROM110. And although it decodes in the instruction decoding circuit 160, the following operation is performed when this instruction is a target instruction at this time. That is, the data stored in the general-purpose register (rs) specified by instruction code are inputted into ALU190 through a data bus 182 from a register file 90. Moreover, the data of a specified register are inputted into ALU190 for an ext instruction just before being held at the EXT register 172. And ALU190 performs the operation shown by the operation code of Type 1 instruction, and stores the result of an operation in the general-purpose register (rd) of a register file 90 through a data bus 192.

[0054] In addition, the process in which an ext instruction holds the data of a specified register to the EXT register 172 is as drawing 2 – drawing 3 having explained.

[0055] Therefore, when Type 1 instruction is executed combining an ext instruction, it has the effectiveness same with having executed 3 operand instructions which have one destination register and two source registers in an operand.

[0056] 5. Explain the example which adds the displacement of the address to the example which adds displacement during an instruction of a target instruction using a prefix instruction, next an operand using a prefix instruction in an instruction (henceforth Type 2 instruction for convenience) of the type which performs using as the address the value stored in the specified register.

[0057] The configuration of instruction code is common in that it has in an operand the appointed field of Type 1 instruction shown in drawing 4, and two registers. Although the contents of the operation



code differ, since there is especially no effect in explanation of the gestalt of this operation, explanation is omitted. The load instruction which is an example of Type 2 instruction is taken and explained to an example. A load instruction is an instruction for loading to the register which shows the data memorized to the address specified by rs to rd.

[0058] First, actuation of CPU in case this load instruction is executed independently is explained. CPU reads the data which make the address the value stored in the general-purpose register (rs) specified by instruction code from RAM320, and stores them in the general-purpose register (rd) specified by instruction code.

[0059] A Type 2 instruction like a load instruction can also be executed independently, and performing combining the last ext instruction is also possible. When it performs combining an ext instruction, with the data of the register specified by the instruction code of an ext instruction, displacement is added and Type 2 instruction is executed.

[0060] The contents of activation when Type 2 instruction is executed combining an ext instruction are explained using drawing 2. First, Type 2 instruction is inputted into an instruction register 150 through the instruction data bus 130 from ROM110. And although it decodes in the instruction decoding circuit 160, when this instruction is a target instruction at this time, the address stored in the general-purpose register (rs) specified by instruction code is taken out from a register file 90, and is inputted into the address-arithmetic machine 340 through the X bus 350.

[0061] Moreover, the data of a specified register are inputted into the address-arithmetic machine 340 for an ext instruction just before being held at the EXT register 172. The address-arithmetic machine 340 adds these two inputs, and generates the address. The address of RAM320 is specified through the data address bus 310 with the generated address, the data stored in this address are inputted into a register file 90 through a data bus 140, and it stores in the general-purpose register (rd) specified by instruction code.

[0062] In addition, in advance of this process, the immediate of an ext instruction is as drawing 2 - drawing 3 having explained the process held EXT register 172.

[0063] Therefore, when Type 2 instruction is executed combining an ext instruction, the contents of operation are extended so that displacement may be added and performed to an operand.

---

[Translation done.]

#### \* NOTICES \*

Japan Patent Office is not responsible for any damages caused by the use of this translation.

1.This document has been translated by computer. So the translation may not reflect the original precisely.

2.\*\*\*\* shows the word which can not be translated.

3.In the drawings, any words are not translated.

---

#### DESCRIPTION OF DRAWINGS

---

[Brief Description of the Drawings]

[Drawing 1] It is the functional block diagram of CPU of the gestalt of operation of this invention.

[Drawing 2] A part required for explanation of the configuration of the semiconductor integrated circuit built in the microcomputer is illustrated.

[Drawing 3] It is the timing-chart Fig. having shown relation with the signal for operation modification which the instruction code inputted and an instruction decoding circuit generate.

[Drawing 4] It is drawing having shown the bit field of the instruction code of Type 1 instruction.

[Drawing 5] It is drawing having shown the bit field of the instruction code of an EXT instruction.

[Description of Notations]

10 Microcomputer

30 CPU

40 Instruction Decoding Section

50 Ext Instruction-Processing Section

60 Instruction-Execution Section

70 Data Operation Part

80 Address-Arithmetic Section

90 Register File

110 ROM

120 Instruction Address Bus

130 Instruction Data Bus

140 Data Bus

150 Instruction Register

160 Instruction Decoding Circuit

161 Control Signal

170 Target Instruction Expansion Circuit

172 EXT Register

190 ALU

310 Data Address Bus

320 RAM

330 Program Counter PC

340 Address-Arithmetic Machine

---

[Translation done.]

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2000-284962

(P2000-284962A)

(43) 公開日 平成12年10月13日 (2000.10.13)

(51) Int.Cl. <sup>7</sup>	識別記号	F I	テーマコード <sup>*</sup> (参考)
G 0 6 F 9/318		G 0 6 F 9/30	3 2 0 B 5 B 0 3 3
9/30	3 1 0		3 1 0 A
	3 5 0		3 5 0 A

審査請求 未請求 請求項の数 3 O L (全 8 頁)

(21) 出願番号 特願平11-94060

(22) 出願日 平成11年3月31日 (1999.3.31)

(71) 出願人 000002369

セイコーエプソン株式会社

東京都新宿区西新宿2丁目4番1号

(72) 発明者 松岡 弘樹

長野県諏訪市大和3丁目3番5号 セイコ

ーエプソン株式会社内

(74) 代理人 100093388

弁理士 鈴木 喜三郎 (外2名)

Fターム(参考) 5B033 AA06 BA05 BB01 BE05 DA02

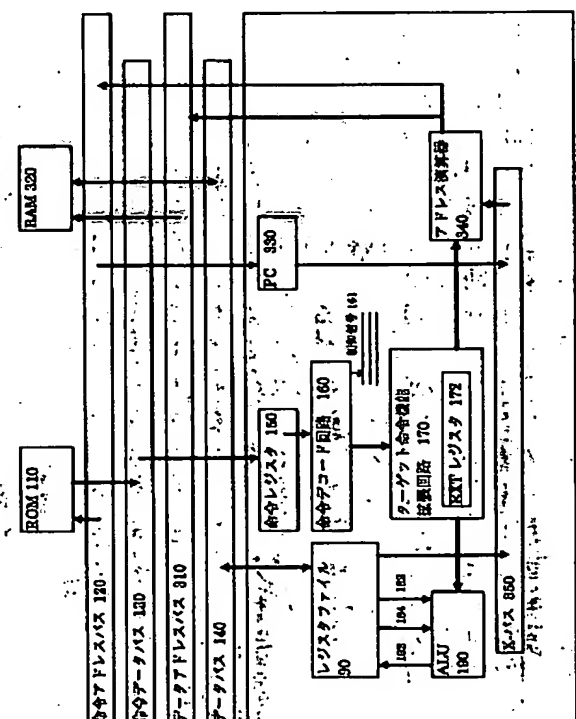
(54) 【発明の名称】 マイクロコンピュータ

(57) 【要約】

【課題】 短い命令コードで記述出来ないオペレーションを実行するために、オペレーション内容を拡張する機能を有するマイクロコンピュータを提供する。

【解決手段】 該マイクロコンピュータは、入力した命令コードを解釈し、該命令を実行するために必要な処理を行うもので、前記命令デコード部40と、レジスタファイル90と、前記命令デコード部40が解析した命令のオペレーション内容に基づき該命令の実行をおこなう命令実行部60を含む。前記命令デコード部40は、プリフィックス命令によりオペレーションを拡張する処理を行うext命令処理部50を含む。

【効果】 プリフィックス命令とターゲット命令を組み合わせることにより幅の広いレジスタをオペランド値として使い、2命令でより広いアドレス空間にアクセスできる。このため命令コードサイズを小さくでき、かつ実行サイクル数も少なくできる。また、命令長が短いタイプのCPUでは困難であったレジスタ間オペランド命令を実現することができる。



(2)

## 【特許請求の範囲】

【請求項1】 命令コード解析手段は、プリフィックス命令入力後にターゲット命令を入力した場合、該ターゲット命令のオペレーション内容を拡張して解釈するターゲット命令機能拡張手段を含み、命令実行手段は、前記ターゲット命令機能拡張手段で拡張されたオペレーション内容で該ターゲット命令を実行することを特徴とするマイクロコンピュータ。

【請求項2】 請求項1記載の、前記ターゲット命令機能拡張手段は、プリフィックス命令入力後に2オペランド命令である所定のターゲット命令を入力した場合、該ターゲット命令の2つのオペランドとプリフィックス命令に含まれた該レジスタの内容を用いて、該ターゲット命令のオペレーション内容を3オペランド命令に拡張して解釈することを特徴とするマイクロコンピュータ。

【請求項3】 請求項1記載の、前記ターゲット命令機能拡張手段は、プリフィックス命令入力後にレジスタ指定値を有する所定のターゲット命令を入力した場合、プリフィックス命令の命令コードに含まれたレジスタ番号のレジスタの内容に基づき、ターゲット命令の命令コードに指定されたレジスタに格納されたアドレスのディスプレースメントを作成し、該ディスプレースメントを用いて実行するようにオペレーション内容を拡張することを特徴とするマイクロコンピュータ。

## 【発明の詳細な説明】

## 【0001】

【発明の属する技術分野】 本発明は、情報処理回路及び半導体集積回路装置、前記情報処理回路を内蔵するマイクロコンピュータ、該マイクロコンピュータを用いて構成された電子機器に関する。

## 【0002】

【従来の技術】 従来、32ビットのデータを処理できるRISC方式のマイクロコンピュータでは、32ビット幅に固定された命令コードを用いられていた。その理由は、固定長ビット幅の命令コードを用いると、可変長ビット幅の命令コードを用いる場合に比べ、命令のデコードに要する時間を短縮でき、また、マイクロコンピュータの回路規模を小さくすることが出来るからである。

【0003】 ところが、32ビットのマイクロコンピュータにおいても、特に32ビットも必要としない命令コードも多い。従って全ての命令の命令コードを32ビットで記述すると、命令コードに冗長な部分が生じる命令が多くなり、メモリの使用効率が悪くなる。

【0004】 この様な場合、冗長な命令コードを論理圧縮しておき、マイクロコンピュータの内部で元の命令にデコードしながら、命令を実行することも可能である。しかし、この様な方式では、制御回路が複雑になるという問題があった。

## 【0005】

【発明が解決しようとする課題】 そこで、本願の発明者

は、制御回路を複雑にすることなくメモリの使用効率を向上させるために、処理出来るデータのビット幅より短いビット幅の固定長命令コードを処理するマイクロコンピュータについての検討を行っていた。しかし、例えば32ビット固定長の命令コードを単に16ビットの固定長にすると以下のような問題点が生じる。例えば、16ビットの命令コードでは、3オペランド命令のオペランドのフィールドを確保することは難しい。従って、この様に短い命令コードでは記述が難しいオペレーションをどの様に処理するのかという問題が生じる。上記問題を解決するためには、短い命令コードで記述出来ないオペレーションを実行するために、オペレーション内容を拡張する機能が必要となる。

【0006】 また、本発明の他の目的は、制御回路を複雑にすることなく、オペレーション内容を拡張する機能を有する情報処理回路、半導体集積回路装置、マイクロコンピュータを提供することである。

## 【0007】

【課題を解決するための手段】 本発明のマイクロコンピュータで処理される命令は、通常の命令と、プリフィックス命令にわけることができる。通常の命令のうち所定の命令については、プリフィックス命令のターゲットとなることができる（プリフィックス命令のターゲットになった命令をターゲット命令という）。プリフィックス命令は、それ単独では命令実行手段において実行されず、後続のターゲット命令が実行される際に、そのターゲット命令の機能を拡張する機能を有する。

【0008】 前記目的を達成するため、請求項1の発明は、所与のターゲット命令と、該ターゲット命令の機能を拡張するためのプリフィックス命令が入力される情報処理回路であって、前記ターゲット命令及びプリフィックス命令を入力し、該命令コードのオペレーション内容を解析する命令コード解析手段と、前記命令コード解析手段が解析したオペレーション内容に基づき命令を実行する命令実行手段とを含むことを特徴とする。

【0009】 命令コード解析手段は、入力した命令コードを解読し、命令の対象となるデータの記憶されている記憶手段の番地を計算する等、命令実行手段が命令を実行するために必要な処理を行う。

【0010】 前記命令コード解析手段は、プリフィックス命令入力後にターゲット命令を入力した場合、該ターゲット命令のオペレーション内容を拡張して解釈するターゲット命令機能拡張手段を含み、前記命令実行手段は、前記ターゲット命令機能拡張手段で拡張されたオペレーション内容で該ターゲット命令を実行することを特徴とする。

【0011】 この様にすると、ターゲット命令をプリフィックス命令と組み合わせて実行させることにより、ターゲット命令に拡張された機能を実現させることができる。このため、命令数を減らすことができ、さらには命

(3)

3

令コードに使用するビット数を減らすことが可能となる。

【0012】また、ビット数の少ない命令コードでは、実現が難しい機能を、プリフィックス命令とターゲット命令を組み合わせることにより実現することも可能となる。このため、命令コードの短縮によるメモリの使用効率を改善することができる。

【0013】請求項2の発明は、請求項1において、前記ターゲット命令機能拡張手段は、プリフィックス命令入力後に2オペランド命令である所定のターゲット命令を入力した場合、該ターゲット命令の2つのオペランドとプリフィックス命令に含まれた番号のレジスタのデータを用いて、該ターゲット命令のオペレーション内容を3オペランド命令に拡張して解釈することを特徴とする。

【0014】この様すると、2オペランド命令のターゲット命令をプリフィックス命令と組み合わせて実行させることにより、該ターゲット命令の操作機能を3オペランド命令に拡張して実行させることができる。従って、2オペランド分記述領域しかない短い命令コードを使用しても3オペランド命令を実現することが出来る。このため、命令数及び命令コードに使用するビット数を減らすことが可能となる。

【0015】請求項3の発明は、請求項1において、前記ターゲット命令機能拡張手段は、プリフィックス命令入力後にレジスタ指定値を有する所定のターゲット命令を入力した場合、プリフィックス命令の命令コードに含まれた番号のレジスタのデータを、ターゲット命令の命令コードに指定されたレジスタに格納されたアドレスのディスプレースメントとして使い実行するようにオペレーション内容を拡張することを特徴とする。

【0016】このようにすると、プリフィックス命令を用いることにより、オペランドのビット数に限定されずにディスプレースメントの付加が可能になり、レジスタのデータ幅のビット分までディスプレースメントを増やせる。

【0017】また、RISC方式のマイクロコンピュータは、固定長の短い命令コードを処理することにより、命令のデコードに要する時間を短縮し、マイクロコンピュータの回路規模を小さくすることを目的としている。従って、本発明によれば、これらの目的を容易に実現出来るRISC方式のマイクロコンピュータを提供することが出来る。

【0018】

【発明の実施の形態】以下、本発明の実施の形態を図面に基づき説明する。

【0019】1. CPUの機能説明

図1は、本発明の実施の形態のCPUの機能ブロック図である。該CPU30は32ビット幅のデータを扱うが16ビットの命令コードを処理するよう構成されてい

4

る。

【0020】本CPU30は、命令デコード部40と、命令実行部60とレジスタファイル90を含む。そしてこのCPUは、16ビットの命令データバス130と、命令データアクセスのための命令アドレスバス120と、32ビットのデータバス140と、データアクセスのためのデータアドレスバス310と、コントロール信号のためのコントロールバス390を介して外部と信号のやり取りを行う。

【0021】前記命令デコード部40は、入力した命令コードを解説し、該命令を実行するために必要な処理を行うのもで、本発明の特徴的な機能であるext命令処理部50を含む。レジスタファイル90は、汎用レジスタR0～R15の16本の汎用レジスタ、プログラムカウンタ(PC)、プロセッサステータスレジスタ(PSR)、スタックポインタ(SP)、算術ローレジスタ(ALR)、算術ハイレジスタ(AHR)等のCPUで使用するレジスタを有している。命令実行部60は、前記命令デコード部40が解析した命令のオペレーション内容に基づき該命令の実行をおこなうもので、データの演算を行うデータ演算部70と、アドレスの演算を行うアドレス演算部80を含む。

【0022】該CPU内部のハードウェアのデータサイズは32ビットであるため、前記命令実行部60のデータ演算部70やアドレス演算部80で行われる算術論理演算や、前記レジスタファイルのレジスタのビット数および、データの転送等はすべて32ビットになる。ところが、前記CPU30が処理する命令コードは16ビットの固定長である。通常、命令コードは該命令の基本的なオペレーションを定めるクラスコード、オペコード、ソースレジスタ(rs)やデスティネーションレジスタ(rd)、即値のオペランドで構成される。このため、16ビットの命令コードを使用する場合、使用できる即値のビット数が小さくなり、また、ソースレジスタが一つしか指定できなくなる。

【0023】そこで、本実施の形態では、ext命令という固定長16ビットの命令を新たに設け、レジスタ番号を指定出来るよう構成している。また、ext命令を用いることにより固定長16ビットという短いビット幅の命令コードでは記述出来ないオペレーションを可能にするよう構成している。ext命令は、それ単独ではCPUにおける実行を何ら行わないが、ターゲット命令の直前におくことにより、そのターゲット命令の機能を拡張する命令であるプリフィックス命令である。なお、該プリフィックス命令による機能拡張の対象となる命令を、ターゲット命令という。ext命令の命令コードの構成は図5に示すように、ビット15からビット13に3ビットのクラス指定領域501と、ビット12からビット8に5ビットのオペコードを指定する領域502と、ビット7からビット0に8ビットのレジスタ指定領

5

域403とを有している。レジスタ指定領域403には、いずれかの汎用レジスタを示すコードが格納されている。

【0024】従って、本実施の形態では、プリフィックス命令で指定する汎用レジスタの値を用いて、後続のターゲット命令のオペレーションを拡張することとを特徴とする。この様なext命令の処理を行うのがext命令処理部50であり、詳細は後述する。

【0025】2. 半導体集積回路の構成

マイクロコンピュータに内蔵されている半導体集積回路を例にとり、本発明の特徴的な機能を実現するための回路構成の一例と動作内容について説明する。

【0026】図2は、マイクロコンピュータに内蔵されている半導体集積回路の構成の説明に必要な部分を図示したものである。該半導体集積回路100は、CPU30と、ROM110、RAM320を含む。

【0027】前記CPU30は、前記ROM110、前記RAM320と、16ビットの命令データバス130、該命令データアクセスのための命令アドレスバス120、32ビットのデータバス140、該データアクセスのためのデータアドレスバス310とを介して、外部と信号のやり取りを行う。

【0028】前記ROM110には、CPU30に実行させる処理を記述した16ビットの命令コードが記憶されており、命令コード記憶手段として機能する。そして、該ROM110に通常の命令の命令コードと同じように、前記プリフィックス命令やターゲットとなる命令の命令コードが記憶される。

【0029】前記RAM320は、実行に使用するデータ等を記憶している。

【0030】前記CPU30は、命令レジスタ150と、命令デコード回路160と、EXTレジスタ172、レジスタファイル90と、ALU190と、プログラムカウンタPC330、アドレス演算器340を含む。

【0031】命令レジスタ150は、前記ROM110から、命令データバス130を介して入力された命令コードを格納する。命令デコード回路160は、命令レジスタ150に格納された命令コードを解釈し、実行に必要な制御信号161を出力する。また、命令コード中の汎用レジスタ値を切り出し、必要に応じてEXTレジスタに保持したりする。アドレス演算器340は加減乗除等の演算を行うが、本具体例では、便宜上加算演算をする機能のみを取り上げ説明する。

【0032】プログラムカウンタPC330には、現在実行中の命令のアドレスが格納されている。

【0033】ALU190は、前記拡張された即値やレジスタファイル90のレジスタに格納された値や前記RAM320のデータに対して、算術演算や論理演算やシフト演算を行う。すなわち、ALU190は図1に示さ

(4)

6

れた命令実行部40のデータ演算部70として機能する。

【0034】ターゲット命令機能拡張回路170は、命令デコード回路160が判断したターゲット命令と命令EXTレジスタ172から該ターゲット命令のオペレーションを拡張する。すなわち、命令デコード回路160とターゲット命令機能拡張回路170とEXTレジスタ172は図1に示された命令デコード部40及びext命令処理部50として機能する。

10 【0035】3. ext命令（プリフィックス命令）を用いたターゲット命令の拡張の具体例

まずプリフィックス命令の命令コードに含まれる汎用レジスタの番号を保持する手段について説明する。手段として、ext命令を命令デコード回路160が検出すると、該当するレジスタのデータをEXTレジスタ172に保存する。

【0036】次に、前記EXTレジスタのデータを使いターゲット命令のオペレーションを変更する手段について説明する。

20 【0037】図3は、入力される命令コード及び前記命令デコード回路が生成する制御信号との関係を示したタイミングチャート図である。

【0038】前記命令デコード回路が生成する前記制御信号はext信号330と、ext\_low信号340とである。図3の310は入力される命令コードを表している。またクロック信号320は、図2において図示しないクロック信号発生器で生成されるか又はクロック入力端子から入力される信号である。該クロック信号520は、CPUにおける各種動作の同期をとるために用いられる。例えば、該クロック信号320の立ち上がりに同期して命令アドレスバス120に命令アドレスが出力される。また、該クロック信号320の1周期（1マシンサイクル）毎に前記命令アドレスに基づいてROM110から命令コードが読み出され、命令レジスタ150に保持される。そして、読み出された命令コードに応じたオペレーションは、1マシンサイクル内で完了する。

30 【0039】ext信号330は、ext命令が入力されたときに'1'となる信号である。すなわち、前記命令デコード回路は、入力された命令がext命令である場合ext信号330を'1'にし、入力された命令が通常の命令である場合ext信号330を'0'にするよう動作する。

40 【0040】ext\_low信号340は、ext信号340の状態に応じて出力される信号である。すなわち、前記命令デコード回路は、ext信号330が'1'である時、次のクロック信号の立ち上がりに同期をとって、ext\_low信号340を'1'にし、その次のクロック信号の立ち上がりに同期をとって、ext\_low信号340を'0'とする。

50

(5)

7

【0041】`ext`信号330が'0'、`ext_low`信号550='0'の場合は、通常の命令を処理している状態であり、この場合には、ターゲット命令拡張回路170は、オペレーションの変更を行わない。

【0042】`ext`信号330が'1'の場合は、CPUがプリフィックス命令である`ext`命令を処理している状態である。この時、`ext`命令の指定するレジスタのデータをEXTレジスタ172に保存する。この場合には、ターゲット命令拡張回路170は、オペレーションの変更を行わない。

【0043】`ext`信号330が'0'で、`ext_low`信号550='1'の場合は、`ext`命令の後にターゲット命令である通常の命令を処理している状態である。この場合には、EXTレジスタ172に保持されている先の`ext`命令の命令コードに含まれていた指定レジスタのデータを使いオペレーションを変更する。

【0044】4. プリフィックス命令によるターゲット命令のオペレーション内容拡張の代表的な実施例次に、本発明のプリフィックス命令によるターゲット命令のオペレーション内容拡張の代表的な実施の形態を、2つ汎用レジスタを用いて演算を行うタイプの命令（便宜上、以下タイプ1命令という）の場合を例に取り説明する。

【0045】まず、タイプ1命令の命令コードの構成について説明する。図5は、該タイプ1命令の命令コードのビットフィールドを示した図である。

【0046】同図に示すように、タイプ1命令の命令コードは、ビット15からビット13に3ビットのクラス指定領域401と、ビット12からビット8に5ビットのオペコードを指定する領域402と、ビット7からビット4に4ビットのレジスタ指定領域403と、ビット3からビット0に4ビットのレジスタ指定領域404とを有している。

【0047】また、前記レジスタ指定領域403、404には、それぞれいずれかの汎用レジスタ（rs）（rd）を示すコードが格納されている。rsとはレジスタソース、rdとはレジスタディスティネーションを意味するものがある。タイプ1命令は、汎用レジスタ（rs）に格納されているデータと汎用レジスタ（rd）に格納されているデータにたいしてオペコードで示される演算を行い、結果を汎用レジスタ（rd）に書き込む動作を行う2オペランド命令である。

【0048】次に、タイプ1命令が単独で実行された場合のCPUの動作について説明する。図2において、まず、タイプ1命令はROM110から命令データバス130を介して命令レジスタ150に入力される。そして、命令デコード回路160で解読され、内容に応じた演算がおこなわれる。タイプ1命令においては、命令コードで指定された汎用レジスタ（rd）に格納されたデータがレジスタファイル90よりデータバス182を介

8

して、ALU190に入力される。また、命令コードで指定された汎用レジスタ（rs）に格納されたデータがレジスタファイル90よりデータバス184を介して、ALU190に入力される。そしてALU190はタイプ1命令のオペコードで示される演算を行い、演算結果をデータバス192を介して、レジスタファイル90の汎用レジスタ（rd）に格納する。

【0049】前記タイプ1命令は単独で実行することも出来るし、直前の`ext`命令と組み合わせて実行することも可能である。

【0050】タイプ1命令を単独で実行する場合及び`ext`命令と組み合わせて実行する場合のオペレーションを表した式は以下の通りである。

【0051】(A)タイプ1命令のオペレーション  

$$rd = rs \text{ op. } rd$$

(B) EXT命令+タイプ1命令のオペレーション  

$$rd = rs \text{ op. EXTレジスタ}$$

式(A)は、タイプ1命令が単独で実行された場合のオペレーションを表した式である。単独で実行される場合は、同式に示すように、タイプ1命令の本来のオペレーションが実行されるため、EXTレジスタのデータは使用されない。なお、タイプ1命令が単独で実行された場合の実行内容については前述した通りである。

【0052】式(B)は、タイプ1命令が直前の`ext`命令と組み合わせて実行された場合のオペレーションを表した式である。同式に示すように、タイプ1命令が直前の`ext`命令と組み合わせて実行された場合は、EXTレジスタのデータと、汎用レジスタ（rs）の内容との演算結果を、汎用レジスタ（rd）に書き込むようにオペレーション内容が拡張する。

【0053】タイプ1命令が`ext`命令と組み合わせて実行された場合の実行内容について説明する。図2において、まず、タイプ1命令はROM110から命令データバス130を介して命令レジスタ150に入力される。そして、命令デコード回路160で解読されるが、このとき該命令がターゲット命令になっている場合は、以下のオペレーションを実行する。すなわち、命令コードで指定された汎用レジスタ（rs）に格納されたデータがレジスタファイル90よりデータバス182を介して、ALU190に入力される。また、EXTレジスタ172に保持されていた直前の`ext`命令が指定したレジスタのデータが、ALU190に入力される。そして、ALU190はタイプ1命令のオペコードで示される演算を行い、演算結果をデータバス192を介して、レジスタファイル90の汎用レジスタ（rd）に格納する。

【0054】なお、`ext`命令が指定したレジスタのデータをEXTレジスタ172に保持する過程は、図2～図3で説明した通りである。

【0055】従って、タイプ1命令を`ext`命令と組み

(6)

9

合わせて実行した場合、オペランドに1つのディスティネーションレジスタと2つのソースレジスタとを有するような3オペランド命令を実行したのと同様の効果を有する。

【0056】5. プリフィックス命令を用いてターゲット命令の命令中にディスプレースメントを付加する実施例

次に、オペランドに指定したレジスタに格納された値をアドレスとして用いて実行を行うタイプの命令（便宜上、以下タイプ2命令という）において、プリフィックス命令を用いてアドレスのディスプレースメントを付加する例を説明する。

【0057】命令コードの構成は図4に示すタイプ1命令と、2個のレジスタの指定領域をオペランドに有する点で共通する。オペコードの内容が異なってくるが、本実施の形態の説明において特に影響は無いので説明を省略する。タイプ2命令の一例であるロード命令を例にとり、説明する。ロード命令は、rsで指定されるアドレスに記憶されているデータをrdに示すレジスタにロードするための命令である。

【0058】まず、該ロード命令が単独で実行される場合のCPUの動作について説明する。CPUは、命令コードで指定された汎用レジスタ(rs)に格納された値をアドレスとするデータをRAM320から読み出し、命令コードで指定された汎用レジスタ(rd)に格納する。

【0059】ロード命令のようなタイプ2命令は、単独で実行することも出来るし、直前のext命令と組み合わせて実行することも可能である。ext命令と組み合わせて実行された場合は、ext命令の命令コードで指定されたレジスタのデータによって、ディスプレースメントが付加されて、タイプ2命令が実行される。

【0060】タイプ2命令がext命令と組み合わせて実行された場合の実行内容について図2を用いて説明する。まず、タイプ2命令はROM110から命令データバス130を介して命令レジスタ150に入力される。そして、命令デコード回路160で解読されるが、このとき該命令がターゲット命令になっている場合は、命令コードで指定された汎用レジスタ(rs)に格納されたアドレスをレジスタファイル90から取り出してXバス350を介して、アドレス演算器340に入力する。

【0061】また、EXTレジスタ172に保持されていた直前のext命令が指定したレジスタのデータが、アドレス演算器340に入力される。アドレス演算器340はこれらの2つの入力を加算してアドレスを生成する。生成されたアドレスによりデータアドレスバス310を介して、RAM320のアドレスを指定し、該アド

10

レスに格納されたデータをデータバス140を介してレジスタファイル90に入力し、命令コードで指定された汎用レジスタ(rd)に格納する。

【0062】なお、この過程に先だってext命令の即値がEXTレジスタ172保持される過程は、図2～図3で説明した通りである。

【0063】従って、タイプ2命令をext命令と組み合わせて実行した場合、オペランドにディスプレースメントを付加して実行するようにオペレーション内容が拡張する。

【図面の簡単な説明】

【図1】本発明の実施の形態のCPUの機能ブロック図である。

【図2】マイクロコンピュータに内蔵されている半導体集積回路の構成の説明に必要な部分を図示したものである。

【図3】入力される命令コード及び命令デコード回路が生成するオペレーション変更用信号との関係を示したタイミングチャート図である。

【図4】タイプ1命令の命令コードのビットフィールドを示した図である。

【図5】EXT命令の命令コードのビットフィールドを示した図である。

【符号の説明】

10 マイクロコンピュータ

30 CPU

40 命令デコード部

50 ext命令処理部

60 命令実行部

70 データ演算部

80 アドレス演算部

90 レジスタファイル

110 ROM

120 命令アドレスバス

130 命令データバス

140 データバス

150 命令レジスタ

160 命令デコード回路

161 制御信号

170 ターゲット命令機能拡張回路

172 EXTレジスタ

190 ALU

310 データアドレスバス

320 RAM

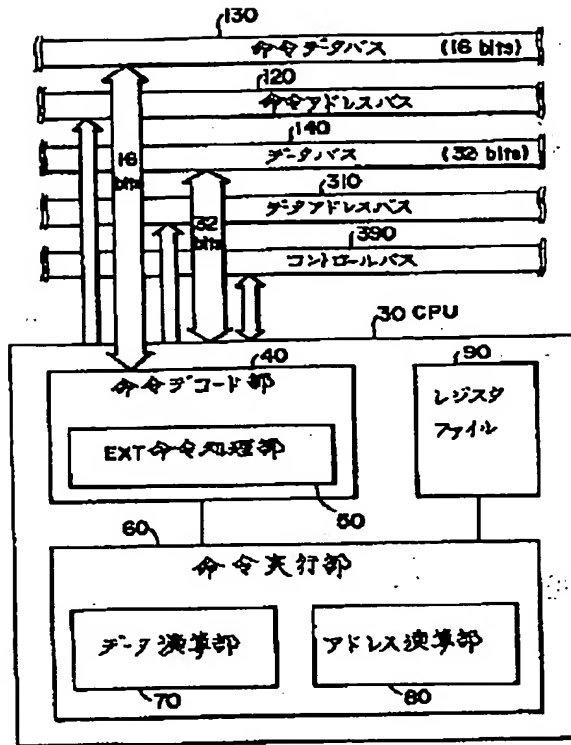
330 プログラムカウンタPC

340 アドレス演算器

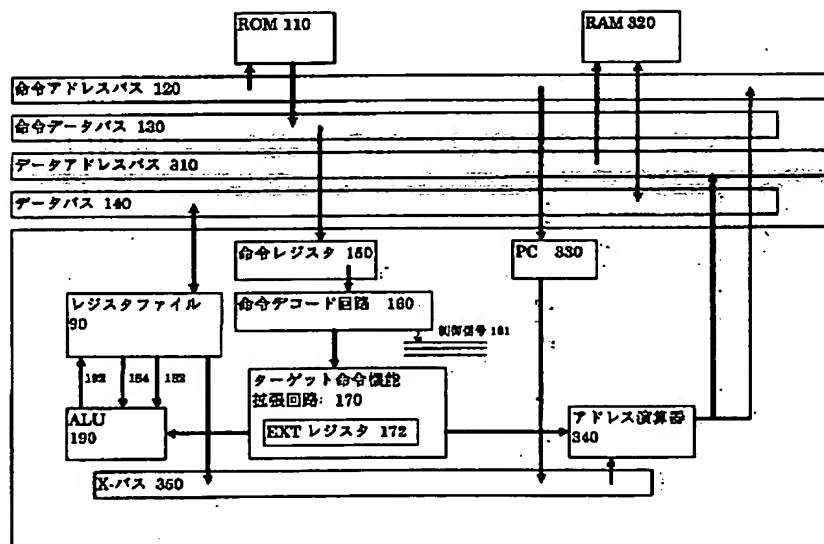


(7)

【図1】

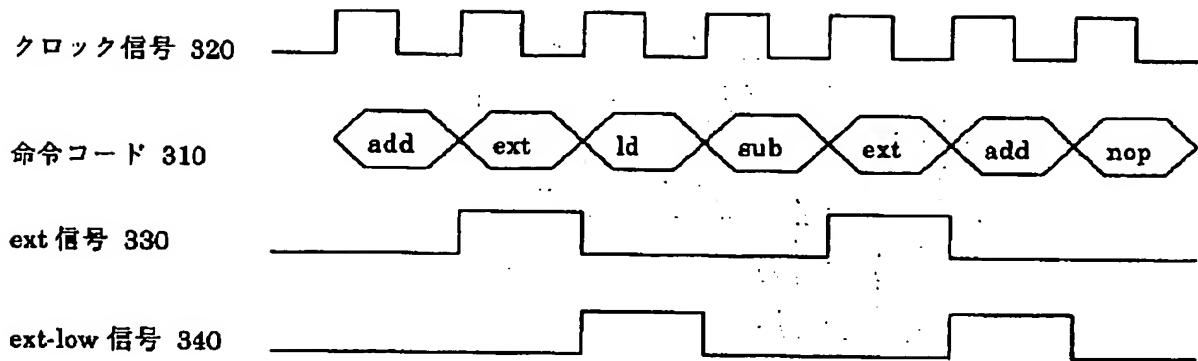


【図2】



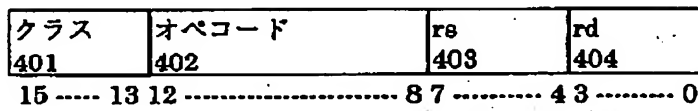
(8)

【図3】



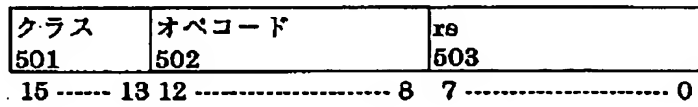
【図4】

タイプ1 命令の命令コードの構成



【図5】

EXT 命令



【公報種別】特許法第17条の2の規定による補正の掲載

【部門区分】第6部門第3区分

【発行日】平成15年7月18日(2003.7.18)

【公開番号】特開2000-284962(P2000-284962A)

【公開日】平成12年10月13日(2000.10.13)

【年通号数】公開特許公報12-2850

【出願番号】特願平11-94060

【国際特許分類第7版】

G06F 9/318  
9/30 310  
350

【FI】

G06F 9/30 320 B  
310 A  
350 A

【手続補正書】

【提出日】平成15年4月8日(2003.4.8)

【手続補正1】

【補正対象書類名】明細書

【補正対象項目名】請求項1

【補正方法】変更

【補正内容】

【請求項1】命令コード解析手段と命令実行手段とを有

するマイクロコンピュータであつて、前記命令コード解析手段は、プリフィックス命令入力後にターゲット命令を入力した場合、該ターゲット命令のオペレーション内容を拡張して解釈するターゲット命令機能拡張手段を含み、命令実行手段は、前記ターゲット命令機能拡張手段で拡張されたオペレーション内容で該ターゲット命令を実行することを特徴とするマイクロコンピュータ。